21152 PCI-to-PCI Bridge Hardware Implementation

Application Note

March 1999

Order Number: 278066-002

Application Note

Contents

Figures

Tables

1.0 Introduction

This document presents guidelines for hardware implementation of the 21152 PCI-to-PCI Bridge chip (21152) in a system. This application note is limited to hardware implementation of the 21152 only and does not cover any devices that might be behind the 21152 or any initialization code needed to configure the 21152 chip.

This application note includes implementation notes on layout, clocking, secondary bus IDSEL mapping, interrupt routing, and secondary bus arbitration. Implementation on both a motherboard and an option card is covered. For most situations, hardware implementation issues are the same. In addition, guidelines for interrupt routing on option cards are provided in the interrupt discussion.

The following related documents may be useful:

- *21152 PCI-to-PCI Bridge Data Shee*t
- *21152 PCI-to-PCI Bridge Configuration Application Note*
- *21152 PCI-to-PCI Bridge Evaluation Board User's Guide*

The following PCI Special Interest Group documents may also be useful:

- *PCI Local Bus Specification, Revision 2.1*
- *PCI-to-PCI Bridge Architecture Specification, Revision 1.0*
- See the Technical Support and Ordering Information section at the end of this document for ordering information.

2.0 Functional Overview

The 21152 fully complies with the *PCI Local Bus Specification, Revision 2.1*. The 21152 has full support for delayed transactions, which enables the buffering of memory read, I/O, and configuration transactions. The 21152 supports buffering of simultaneous multiple posted write and delayed transactions in both directions.

The 21152 provides a connection between two independent PCI buses. The two PCI buses are referred to as the primary PCI bus, which is the PCI bus closest to the host CPU, and the secondary PCI bus, which is the PCI bus farthest from the host CPU.

The 21152 allows the two PCI buses to operate concurrently. A master and target on the same PCI bus can communicate while the other PCI bus is busy.

The 21152 has two common applications. System designers can use the 21152 on a motherboard to add more devices or add-in card slots than a single PCI bus can support. Add-in card designers can use the 21152 to enable multiple devices on a single option card. Up to four bus mastering devices can be attached to the secondary bus of the 21152.

The 21152 supports both 5 V and 3.3 V signaling environments.

The 21152 PCI-to-PCI Bridge has the following features:

- Complies fully with Revision 2.1 of the *PCI Local Bus Specification, Revision 2.1*
- Complies fully with Revision 1.0 of the *PCI-to-PCI Bridge Architecture Specification, Revision 1.0*
- Implements delayed transactions for all PCI configuration, I/O, and memory read commands–up to three transactions simultaneously in each direction
- Allows 88 bytes of buffering (data and address) for posted memory write commands in each direction–up to 5 posted write transactions simultaneously in each direction
- Allows 72 bytes of read data buffering in each direction
- Provides concurrent primary and secondary bus operation to isolate traffic
- Provides five secondary clock outputs:
  - — Low skew, permitting direct drive of option slots
  - — Individual clock control through configuration space
- Provides arbitration support for four secondary bus devices:
  - — A programmable 2-level arbiter
  - — Hardware disable control, permitting use of an external arbiter
- Provides enhanced address decoding:
  - — A 32-bit I/O address range
  - — A 32-bit memory-mapped address range
  - — A 64-bit prefetchable memory address range
  - — ISA-aware mode for legacy support in the first 64 KB of I/O address range
  - — VGA addressing and VGA palette snooping support
- Supports PCI transaction forwarding for the following commands:
  - — All I/O and memory commands
  - — Type 1 to Type 1 configuration commands
  - — Type 1 to Type 0 configuration commands (downstream only)
  - — All Type 1 to special cycle configuration commands
- Includes downstream lock support
- Supports both 5-V and 3.3-V signaling environments

The 21152 makes it possible to extend a system’s load capability limit beyond that of a single PCI bus by allowing motherboard designers to add more PCI devices, or more PCI option card slots, than a single PCI bus can support.

3.0 21152 Power Supply

The 21152 must be powered by a 3.3-V power supply. If 3.3-V is not available in the system, it can be generated from a 5-V power supply by using a voltage regulator. Figure 1 shows a recommended regulator circuit. A tantalum electrolytic capacitor of at least 10 µF is required at the output of the regulator.

Figure 1. 21152 Voltage Regulator Circuit

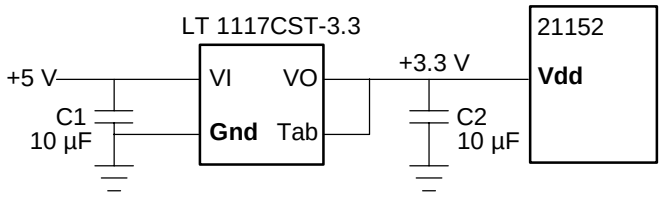


Table 1 lists the voltage regulators that can be used. To implement any of these regulators, refer to the vendor's data sheet.

Table 1. Voltage Regulator Vendors

Vendor	Part Number
Linear Technology	LT1117CST-3.3
Texas Instruments	TLV2217-33
Motorola	MC33269D
National Semiconductor	LM3940

3.1 Power Sequencing

Early PCI applications used a single 5V power supply. As 3.3V PCI devices became available, a combination of 3.3V and 5V supplies were needed for a single option. Most, if not all, of these applications generally use a voltage regulator circuit as part of application to supply 3.3V from the 5-volt source.

In the natural circuit timing of a voltage regulator, the 5V supply precedes the 3.3V supply by default. When a system provides 3.3-volts from a separate power supply the timing of the power supplies when turning on or ramping down may not be well defined. It is important that the two supplies should follow the example of the voltage regulator circuit (see Figure 1).

For the PCI-to-PCI bridge products, there is a specified sequence requirement for the 3.3V and 5V supply activation and deactivation. These power sequencing requirements for the 21150, 21152, 21153, 21154, and 21554 products is as follows (Vdd refers to 3.3V and Vcc refers to 5V):

While activating or deactivating power the 5-volt and 3.3-volt supplies should track each other within 1.8-volts:

- If Vdd < 3.0V then Vcc - Vdd < 1.8V
- If Vcc < 1.8V then Vdd can be 0
- If Vcc > 1.8V then Vdd must be > 0

The 3.3V (Vdd) supply may lag the 5V (Vcc) supply by up to 1.8 volts while the supplies are changing. When both supplies have settled to their final values, Vcc can be up to 5.25 volts if Vdd > 3.0V.

Note: Slow supply ramp rates, greater than 10 ms, could cause die heating in CMOS devices. This would depend on factors other than the I/O of the PCI-to-PCI bridge. Measurements to determine ramp rates should be taken between Vmax and Vmin. It should also be noted that a power down rate greater than 10ms will not damage the device if the Vcc and Vdd requirements that follow are met.

There is no ramp rate (timing to voltage stable) limitation. Only the difference in voltages is important. This restriction applies both to powering up and down. There are no restrictions if the 3.3V supply comes up before 5V, or shuts off last.

Note: To prevent possible damage when using separate 3.3 and 5V power supplies a 1K ohm resistor should be placed between p_vio and s_vio and the 5V supply when 5-volt options are used.

3.2 5 V and 3.3 V Signaling

The 21152 I/O pads are 5 V tolerant and will operate under both 3.3-V and 5-V signaling environments. The primary and secondary PCI buses can be independently interfaced to either a 3.3-V or 5-V signaling environment by connecting the **p_vio** and **s_vio** pins to the appropriate voltages. These pins are connected to a clamp circuit in the pad driver that turns on when the output voltage matches the voltage on the **p_vio** and **s_vio** pins.

Pay special attention to board layout and signal environments when mixing 3.3-V and 5-V devices on the same bus. It is guaranteed by design that the 21152 is reliable in both 3.3-V and 5-V environments when **p_vio** and **s_vio** are connected in accordance with Table 2. In all applications, follow the *PCI Local Bus Specification, Revision 2.1* concerning layout and signal integrity issues.

Table 2. p_vio and s_vio Connections

Signaling Environment Primary Bus	Secondary Bus	p_vio (Pin 24)	s_vio (Pin 35)
5 V	5 V	5 V	5 V
5 V	3.3 V	5 V	3.3 V
3.3 V	5 V	3.3 V	5 V
3.3 V	3.3 V	3.3 V	3.3 V

4.0 Clocking

The following sections provide an overview of 21152 clocking requirements, and explain how to implement clocks when using the 21152 on a motherboard and on an option card.

4.1 21152 Clocking Domains

The 21152 has two clocking domains: one for the primary PCI interface and one for the secondary PCI interface. Each PCI interface has a separate clock input. The primary interface is controlled by the primary clock input, **p_clk**, and the secondary interface and arbiter are controlled by the secondary clock input, **s_clk**.

Both interfaces must operate at the same frequency and must be synchronous to each other. Their edge relationships to each other are well-defined. The relationship between the **p_clk** and **s_clk** inputs has the following restrictions:

- The 21152 operates at a maximum frequency of 33 MHz, and **s_clk** always operates at the same frequency as **p_clk**.
- The maximum delay between **p_clk** and **s_clk** is 7 ns for both rising and falling edges.
- The minimum delay between **p_clk** and **s_clk** is 0 ns, that is, **s_clk** cannot precede **p_clk** for both rising and falling edges.
- To compensate for the 21152 clock buffer characteristics to the secondary bus logic, the rise and fall times of **p_clk** must be constrained so that the clock provided to a slot has a duty cycle derived from maximum rise and fall times minus the clock skew.

  No duty cycle compensation is needed if:

  Tcycle ≥ Tcycle_min + 2 x Tskew

  if Tskew = 750 ps and Tcycle ≥ 31.5 ns (or 31.75 MHz and below)

  Motherboard designers must compensate for duty cycle skew of the 21152 clock buffer by constraining Tr and Tf (rise and fall time) when:

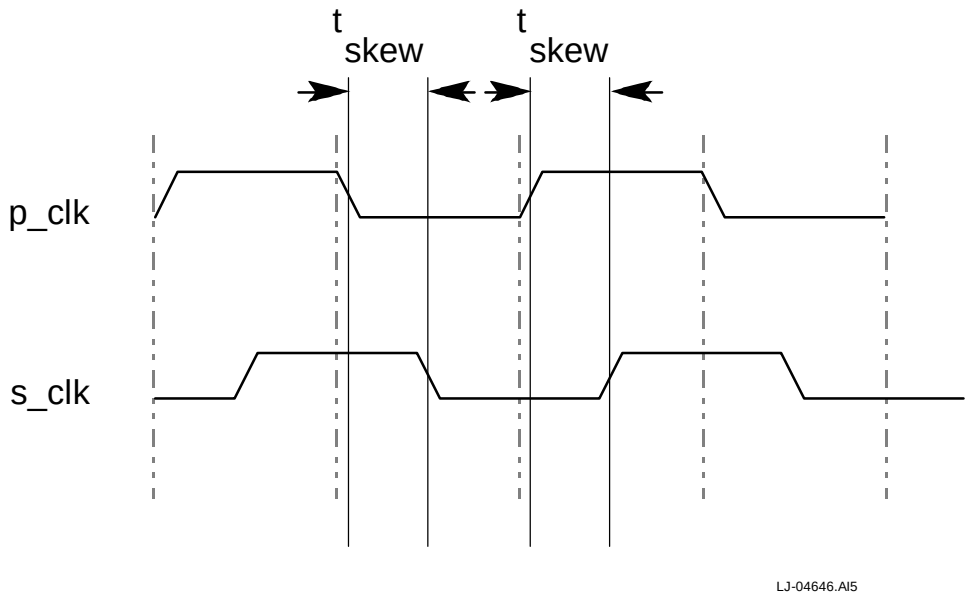
  Tcycle equal to or less than Tcycle_min + 2 x Tskew

  if Tskew = 750 ps and Tcycle equal to or less 31.5 ns (or 31.75 MHz and above)

  Therefore, a Tr and Tf maximum = 3 ns – skew of 750 ps = 2.25 ns will give a Thigh or Tlow of 12.75 ns for a Tcycle minimum of 30 ns.

Figure 2 shows the timing relationship between the primary and the secondary clock inputs.

Figure 2. p_clk and s_clk Relative Timing



4.2 21152 Output Clocks

The 21152 generates five secondary bus clock outputs, **s_clk_o<4:0>**. These clocks are derived from **p_clk**. Use one of the secondary clock outputs as the secondary clock input to the 21152; use the other four outputs for devices on the secondary bus.

The **s_clk_o<4:0>** clock outputs are generated directly from **p_clk** by internal buffers, and their relationships to **p_clk** have the following properties:

- All clock outputs operate at the same frequency as **p_clk**.
- The maximum delay between **p_clk** and **s_clk_o** is 5 ns for both rising and falling edges. Therefore, to meet the **p_clk** and **s_clk** requirements, no more than 2 ns of delay is allowed for secondary clock etch returning to the device secondary clock input.
- The minimum delay between **p_clk** and **s_clk_o** is 0 ns for both rising and falling edges.
- The maximum skew between **s_clk_o** edges is 500 ps.

4.3 Guidelines for Using Secondary Clock Outputs

The guidelines for using secondary clocks are:

- Each secondary clock output is limited to one load.
- One of the secondary clock outputs must be used for the 21152 **s_clk** input.
- Intel recommends using an equivalent amount of etch on the board for all secondary clocks to minimize skew between them.
- Route **s_clk_o<4:0>** using the same length of etch for each output, including that used as **s_clk** input. Intel recommends that the amount of delay due to etch be limited to less than 2 ns (10 inches if etch has a delay of 200 ps/in).
- Intel recommends that you use a series-termination resistor to terminate the **s_clk_o<4:0>** outputs to limit reflections on the clock lines. The value of the resistor depends on the transmission line impedance of the etch, which depends on the board layout. As a result, it is difficult to recommend a precise value. Based on a 75-Ω transmission line, a 50-Ω series resistor has been seen to provide acceptable clock waveforms.

4.4 Clock Implementation on the Motherboard

When the 21152 is implemented on a motherboard and the secondary bus is routed to one or more PCI expansion slots or to another 21152, you can use the 21152 secondary clock outputs **s_clk_o<4:0>**, provided you guarantee that the output clocks meet PCI duty cycle specifications. You must include the 750 ps skew added by the 21152 to the duty cycle in the analysis of the clocks' circuits.

If you choose not to use the secondary clock outputs, you can substitute a low-skew clock buffer to generate secondary clocks.

Table 3 lists some of the buffers that can be used.

Table 3. Low-Skew Clock Buffers

Vendor	Part Number
Texas Instruments	CDC328A
National Semiconductor	CGS74B2525
IDC	1DT74FCT805CT

One of the secondary clock signals must be connected to the 21152 secondary clock input, **s_clk**. Plan the layout of clocks to minimize skew between the 21152 and other PCI devices.

5.0 General Layout Guidelines

When using the 21152, you need to consult the general layout guidelines provided in the *PCI Local Bus Specification, Revision 2.1*.

Clock routing has some special requirements. Guidelines and requirements for clock routing are discussed in the Clocking section of this application note. Guidelines for routing of secondary IDSEL signals are given in the Secondary IDSEL Mapping section of this application note.

This section discusses pull-ups and expansion card routing.

5.1 Motherboard Requirements

Consult the physical requirements in the *PCI Local Bus Specification, Revision 2.1* for the layout of the PCI on a system motherboard. The system timing requirements, clock skew, signal velocity, and round-trip propagation delay of 10 ns should be the goal for operation at 33 MHz.

5.2 Expansion Card Routing

Follow the guidelines and requirements for routing on expansion cards in the *PCI Local Bus Specification Revision 2.1*. This section highlights some important requirements.

PCI signals coming from the motherboard onto the expansion card must be limited to only one load. This includes the primary clock. These are the signals on the primary interface of the 21152. These signals also have trace length limitations, which are 1.5 inches for PCI signals and 2.5 inches for the primary clock.

PCI signals on the secondary side of the 21152 do not need to adhere to these restrictive loading and trace length requirements. The secondary PCI bus can support the full 10 loads (including the 21152) and can be treated like a motherboard PCI bus.

5.3 Pull-Ups

Pull-up resistors are required on the following shared PCI control signals: FRAME#, IRDY#, TRDY#, STOP#, DEVSEL#, PERR#, SERR#, and LOCK#. These signals must have pull-ups on both the primary and secondary PCI buses, and they should be pulled up to **p_vio** or **s_vio** voltages. When optional shared signals such as LOCK# are not used, they must be pulled up for proper 21152 operation.

For resistor values, use the formula specified in Section 4.3.3 of the *PCI Local Bus Specification, Revision 2.1*. In this formula, use a value of 1.5 mA for Iol (per the specifications shown in the *21152 PCI-to-PCI Bridge Data Sheet*) and the appropriate value for **Vcc**.

When an external arbiter is used, or if all of the **s_req_l** lines are not used, all of the unused **s_req_l** lines should have pull-up resistors.

If the 21152 is implemented on a motherboard, both primary and secondary bus pull-ups should be located on the motherboard. When the 21152 is being used on an option card with the primary bus interfacing to the card edge, primary bus pull-ups must be located on the motherboard and secondary bus pull-ups must be located on the option card. In addition, **s_req_l<3:0>** must be pulled up to the **s_vio** voltage. The recommended value for the pull-up resistors is 5 KΩ.

5.4 Decoupling

According to Section 4.4.2.1 of the *PCI Local Bus Specification, Revision 2.1:*

"Under typical conditions, the **Vcc** plane to ground plane capacitance will provide adequate decoupling for the **Vcc** connector pins. The maximum trace length from a connector pad to the **Vcc**/**Gnd** plane is 0.25 inches, assuming a 20 mil trace width.

However, on the Universal board, it is likely that the I/O buffer power rail will not provide adequate capacitance to the ground plane to provide the necessary decoupling. Pins labeled "+V i/o" should be decoupled to ground with an average of 0.047 µF per pin.

Additionally, all 3.3 V pins (even if they are not actually delivering power), and any unused 5-V and V i/o pins on the PCI edge connector provide an ac return path. These pins must have plated edge fingers and be decoupled to the ground plane on the add-in board to ensure they continue to function as efficient ac reference points:

- The decoupling must average at least 0.01 µF (high-speed) per **Vcc** pin.
- The trace length from pin pad to capacitor pad shall be no greater than 0.25 inches using a trace width of at least 0.02 inches.
- There is no limit to the number of pins that can share the same capacitor provided that the previous two requirements are met."

5.5 FCC Considerations

To minimize electromagnetic interference (EMI) and to control signal integrity characteristics, follow the suggestions in Section 4.4.3 of the *PCI Local Bus Specification, Revision 2.1*. This section covers trace lengths and routing of signals. Although not an optimal solution where EMI and FCC compliance are concerned, four-layer boards are recommended as a midrange solution. When four-layer boards are used, Intel recommends that you use the following layout guidelines, per the *PCI Local Bus Specification, Revision 2.1*:

"…arrange the signal level layouts so that no high speed signal (e.g., 33 MHz) is referenced to both planes. Signal traces should either remain entirely over the 3.3-V plane or entirely over the 5-V plane. Signals that must cross from one domain to the other should be routed on the opposite side of the board so that they are referenced to the ground plane which is not split. If this is not possible, and signals must be routed over the plane split, the two planes should be capacitively tied together (5-V plane decoupled directly to 3.3-V plane) with 0.01 µF high-speed capacitors for each four signals crossing the split and the capacitor should be placed not more than 0.25 inches from the point the signals cross the split."

5.6 Additional Board Layout Guidelines

The following list contains some additional board layout guidelines:

1. Avoid signals crossing over a split in the ground, power, or both because they will contribute to noise problems.
2. Decoupling for high frequencies:
   a. Add one high-frequency decoupling capacitor per power pin where possible. To minimize inductance, Intel recommends using 0805 or 1206 style surface-mount 0.001 µF capacitors.
   b. The maximum trace length from a connect pad to the **Vcc**/**Gnd** plane is 0.25 inches, assuming a 0.02-inch trace width.

   There is no limit to the number of pins that can share the same capacitor provided that the previous two requirements are met.

   Locate each capacitor as close to the pin as possible. Any etch length that is added at this path is inductive and will cause oscillations. A preferred method of adding decoupling capacitors when the board is crowded is to extend and merge the $\mathbf{V_{DD}}$ pads of the device with the capacitor pad and put a power via in the pad. Form a similar connection for ground.

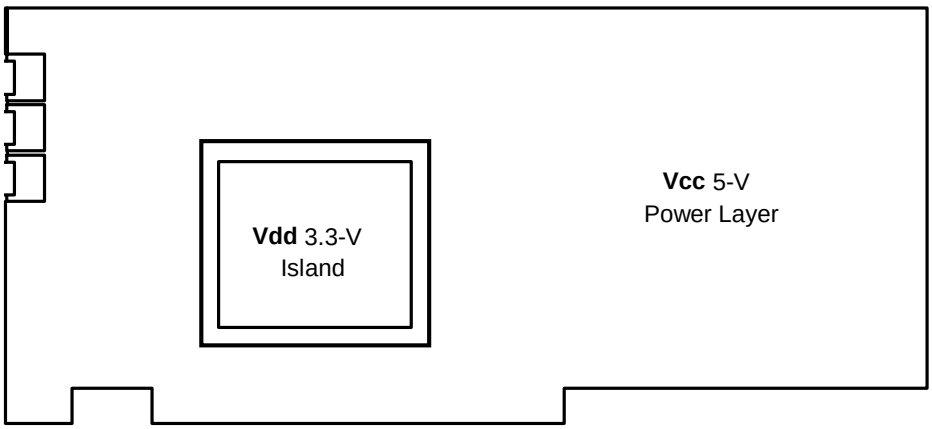
   For effective decoupling, it may be necessary to place components on the reverse side of the board.

   On the Universal board, it is likely that the I/O buffer power will not provide adequate capacitance to the ground plane to provide the necessary decoupling. Pins labeled "+V i/o" (**v_pio** and **v_sio**) should be decoupled to ground with an average of 0.047 µF per pin.

   Additionally, all 3.3-V pins (even if they are not actually delivering power) and any unused 5 V and V I/O pins on the PCI edge connector provide an ac return path. These pins must have plated edge fingers and must be decoupled to the ground plane on the add-in board to ensure that they continue to function as efficient ac reference points.
3. Add one bulk decoupling capacitor per device. The size of the bulk decoupling capacitor should be greater than the total capacitance being charged and discharged. Intel recommends using a 22 µF 1812 style surface-mount capacitor.
4. On six-layer boards and higher, assign power and ground layer as close as possible. This will provide a large decoupling capacitance and will have greater power filtering effect.
5. Bury clocks on internal signal layers. Add a guard signal adjacent to each clock line and terminate the guard signal at both ends.
6. To prevent noise generated by the 21152 from coupling into other components, form a voltage island around the chip. This can be done easily for the chip that requires a voltage supply other than 5 V.

Figure 3 shows Intel Semiconductor's implementation of a voltage island.

Figure 3. 3.3-Voltage Island



7. Logic ground of the board should never be directly connected to chassis ground. Normally, chassis ground will be referenced to the main power supply in the system. The bracket of the option card will be connected to that ground through the chassis.

6.0 Secondary IDSEL Mapping

The *PCI Local Bus Specification, Revision 2.1* defines two formats for configuration transactions in hierarchical systems:

- Type 0 configuration transactions are used to configure devices on the same PCI bus.
- Type 1 configuration transactions are used to configure devices that reside on a downstream PCI bus.

When the 21152 detects a Type 1 configuration transaction for a device connected to the secondary interface of the bridge, the 21152 translates the Type 1 transaction into a Type 0 transaction on the downstream interface.

Instead of using IDSEL to identify the target of a configuration transaction, the Type 1 configuration format uses a 5-bit field at bits <15:11> in the address as a device number. A device number in the Type 1 format is translated by the 21152 into an IDSEL line for Type 0 transactions on the target interface.

As required by the *PCI-to-PCI Bridge Architecture Specification*, the 21152 uses **s_ad<31:16>** as secondary IDSEL lines. The 21152 uses the mapping shown in Table 4 for translation of a device number to an **s_ad** pin.

The 21152 supports mapping for device numbers 0h through Fh. Configuration transactions with device numbers outside of this range are still forwarded, but no IDSEL mapping is performed (**s_ad<31:16>** are 00h).

Table 4. Device Number to Secondary IDSEL Mapping

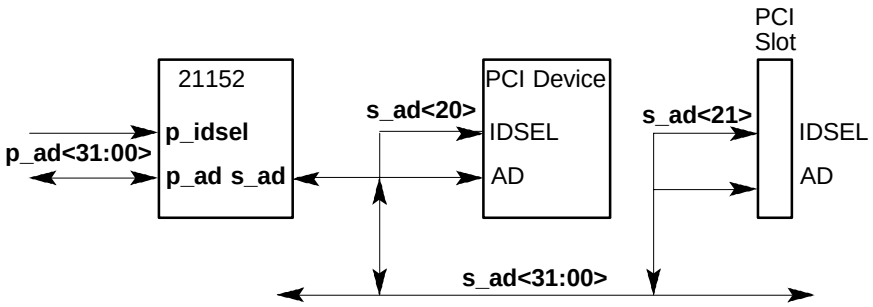
Device Number	p_ad<15:11>	Secondary IDSEL s_ad<31:16>	s_ad Bit
0h	00000	0000 0000 0000 0001	16
1h	00001	0000 0000 0000 0010	17
2h	00010	0000 0000 0000 0100	18
3h	00011	0000 0000 0000 1000	19
4h	00100	0000 0000 0001 0000	20
5h	00101	0000 0000 0010 0000	21
6h	00110	0000 0000 0100 0000	22
7h	00111	0000 0000 1000 0000	23
8h	01000	0000 0001 0000 0000	24
9h	01001	0000 0010 0000 0000	25
Ah	01010	0000 0100 0000 0000	26
Bh	01011	0000 1000 0000 0000	27
Ch	01100	0001 0000 0000 0000	28
Dh	01101	0010 0000 0000 0000	29
Eh	01110	0100 0000 0000 0000	30
Fh	01111	1000 0000 0000 0000	31
10h–1Eh	10000–11110	0000 0000 0000 0000	–
1Fh	11111	Generate special cycle (**p_ad<7:2>**=00h) 0000 0000 0000 0000 (**p_ad<7:2>**≠(00h)	–

When implementing the 21152 in a system, each PCI device and PCI slot connected to the secondary PCI bus must have its IDSEL line connected to one of the **s_ad<31:16>** lines, as shown in Figure 4.

When you assign secondary AD signals to secondary IDSEL signals on an expansion card, make sure that the device numbers are consistent with the required interrupt binding given in Table 5.

Note: Some early PCI host bridges automatically claim all configuration commands directed to device 0. Therefore, Intel recommends that you avoid using device 0.

Figure 4. Secondary IDSEL Implementation Example



Note: Previous versions of Intel Semiconductor's PCI-to-PCI bridge chips that implemented address stepping required a 1-KΩ resistor between the AD# input and the IDSEL# input of the PCI device on the secondary bus. This is no longer a requirement.

7.0 Interrupts

The following section describes how interrupts should be implemented in a 21152 application.

7.1 Data Synchronization and Interrupts

The *PCI Local Bus Specification, Revision 2.1* requires that either the interrupt handler (service routine) or the device that initiates the interrupt guarantees that all buffers are flushed between the device and the final destination. To accomplish this, the interrupt service routine of the device driver can perform a read of the device, or the device itself can perform a read of the last location written by the device. In either case, the read transaction forces buffers between the device and the final destination to be flushed.

Section 6.3.4 of the *PCI Local Bus Specification, Revision 2.1* states, "Device drivers are ultimately responsible for guaranteeing consistency of interrupts and data."

Interrupts originating from secondary bus devices are not routed through the 21152.

7.2 Interrupt Binding on Option Cards

When PCI-to-PCI bridges are used in option cards, a binding is required by the *PCI-to-PCI Bridge Architecture Specification, Revision 1.0* between the device number (as given in the Type 1 configuration address and, therefore, the IDSEL line) and the INTx# line it uses when requesting an interrupt.

The PCI connector has only four interrupt lines assigned to it: INTA#, INTB#, INTC#, and INTD#. Multiple devices might have to share these lines. Refer to Table 5.

Because only the BIOS knows how the PCI INTx# lines are routed to the system interrupt controller, a mechanism is required to inform the device driver which IRQ its device will request an interrupt on. The interrupt line register stores this information.

The BIOS code assumes the binding is as listed in Table 5 behind a PCI-to-PCI bridge and writes the IRQ number in each device. The interrupt binding defined in Table 5 is mandatory for option cards using PCI-to-PCI bridges.

Table 5. Interrupt Binding on Option Cards

Device Number on Secondary Bus	Interrupt Pin on Device	Interrupt Pin on Connector
0, 4, 8, 12, 16, 20, 24, 28	INTA# INTB# INTC# INTD#	INTA# INTB# INTC# INTD#
1, 5, 9, 13, 17, 21, 25, 29	INTA# INTB# INTC# INTD#	INTB# INTC# INTD# INTA#
2, 6, 10, 14, 18, 22, 26, 30	INTA# INTB# INTC# INTD#	INTC# INTD# INTA# INTB#
3, 7, 11, 15, 19, 23, 27, 31	INTA# INTB# INTC# INTD#	INTD# INTA# INTB# INTC#

8.0 Arbitration

The 21152 must arbitrate for use of the primary bus when forwarding upstream transactions, and for use of the secondary bus when forwarding downstream transactions. The arbiter for the primary bus resides external to the 21152, typically on the motherboard.

For the secondary PCI bus, the 21152 implements an internal arbiter. This arbiter can be disabled, and an external arbiter can be used instead.

The following sections describe primary and secondary bus arbitration.

8.1 Primary Bus Arbitration

The 21152 implements a request output pin, **p_req_l**, and a grant input pin, **p_gnt_l**, for primary PCI bus arbitration. The 21152 asserts **p_req_l** when forwarding transactions upstream; that is, it acts as initiator on the primary PCI bus.

As long as at least one pending transaction resides in the queues in the upstream direction, either posted write data or delayed transaction requests, the 21152 keeps **p_req_l** asserted. However, if a target retry, a target disconnect, or a target abort is received in response to a transaction initiated by the 21152 on the primary PCI bus, the 21152 deasserts **p_req_l** for two PCI clock cycles.

For additional information on primary bus cycle arbitration, refer to the *21152 PCI-to-PCI Bridge Data Sheet*.

8.2 Secondary Bus Arbitration

The 21152 implements an internal secondary PCI bus arbiter. This arbiter supports four external masters in addition to the 21152. The internal arbiter can be disabled, and an external arbiter can be used instead for secondary bus arbitration.

8.2.1 Secondary Bus Arbitration Using the Internal Arbiter

To use the 21152 secondary bus arbiter:

- Tie the enable pin, **s_cfn_l**, to ground (low) through a 1-KΩ resistor.
- Connect the four secondary bus request input pins, **s_req_l<3:0>**, and the four secondary bus grant output pins, **s_gnt_l<3:0>**, to external secondary bus masters as required.

Note: The 21152 secondary bus request and grant signals are connected internally to the arbiter and are not brought out to external pins when **s_cfn_l** is low.

The internal secondary arbiter supports a programmable two-level rotating algorithm. Two groups of masters are assigned, a high-priority group and a low-priority group. The low-priority group as a whole represents one entry in the high-priority group; that is, if the high-priority group consists of *n* masters, then in at least every *n*+1 transaction the highest priority is assigned to the low-priority group. Priority rotates evenly among the low-priority group.

If the 21152 detects that an initiator has failed to assert **s_frame_l** after 16 cycles of both grant assertion and a secondary idle bus condition, the arbiter deasserts the grant. That master does not receive any more grants until it deasserts its request for a least one PCI clock cycle.

To prevent bus contention, if the secondary PCI bus is idle, the arbiter never asserts one grant signal in the same PCI cycle in which it deasserts another. It deasserts one grant, and then asserts the next grant, no earlier than one PCI clock cycle later. If the secondary PCI bus is busy, that is, if either **s_frame_l** or **s_irdy_l** is asserted, the arbiter can deassert one grant and assert another grant during the same PCI clock cycle.

8.2.2 Secondary Bus Arbitration Using an External Arbiter

To use an external arbiter, tie the central function enable pin, **s_cfn_l**, high through a 1-KΩ resistor. This disables the internal arbiter.

With **s_cfn_l** tied high, the 21152 reconfigures two pins to be external request and grant pins. The **s_gnt_l<0>** pin is reconfigured to be the 21152's external request pin because it is an output. Connect this pin to one of the request lines of the external arbiter. The **s_req_l<0>** pin is reconfigured to be the external grant pin because it is an input. Connect this pin to one of the grant lines of the external arbiter.

The unused secondary bus grant outputs, **s_gnt_l<3:1>**, are driven high. Unused secondary bus request inputs, **s_req_l<3:1>**, should be pulled high using 1-KΩ resistors.

8.2.3 Bus Parking

Bus parking refers to driving the AD, C/BE#, and PAR lines to a known value while the bus is idle. In general, the device implementing the bus arbiter is responsible for parking the bus or assigning another device to park the bus. A device parks the bus when the bus is idle, its bus grant is asserted, and the device's request is not asserted. The AD and C/BE# signals should be driven first, with the PAR signal driven one cycle later.

The 21152 parks the primary bus only when **p_gnt_l** is asserted, **p_req_l** is deasserted, and the primary PCI bus is idle. When **p_gnt_l** is deasserted, the 21152 tristates the **p_ad**, **p_cbe_l**, and **p_par** signals on the next PCI clock cycle. If the 21152 is parking the primary PCI bus and wants to initiate a transaction on that bus, then the 21152 can start the transaction on the next PCI clock cycle by asserting **p_frame_l** if **p_gnt_l** is still asserted.

If the internal secondary bus arbiter is enabled, the secondary bus is always parked at the last master that used the PCI bus. That is, the 21152 keeps the secondary bus grant asserted to a particular master until a new secondary bus request comes along. After reset, the 21152 parks the secondary bus at itself until transactions start occurring on the secondary bus. If the internal arbiter is disabled, the 21152 parks the secondary bus only when the reconfigured grant signal, **s_req_l<0>,** is asserted and the secondary bus is idle.

Support, Products, and Documentation

If you need general information or support, call **1-800-628-8686** or visit Intel's website at:

http://www.intel.com

Copies of documents that have an ordering number and are referenced in this document, a product catalog, or other Intel literature may be obtained by calling **1-800-548-4725** or by visiting Intel's website for developers at:

http://developer.intel.com